DOCUMENT RESUME

ED 427 677 IR 019 196

AUTHOR Westhead, Martin D.
TITLE EPIC: Building a Structured Learning Environment.
PUB DATE 1996-10-00
NOTE 7p.; In: WebNet 96 Conference Proceedings (San Francisco, CA, October 15-19, 1996); see IR 019 168.
AVAILABLE FROM Web site: http://aace.virginia.edu/aace/conf/webnet/html/369.htm
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Cognitive Structures; Computer Assisted Instruction; Computer Science Education; *Computer Software Development; *Courseware; Distance Education; *Educational Environment; Educational Technology; Foreign Countries; Higher Education; *Instructional Design; Learning Processes; Organization; World Wide Web
IDENTIFIERS Interactive Systems; *Learning Environments; *Structured Learning; University of Edinburgh (Scotland)

ABSTRACT

This paper outlines work in progress at the University of Edinburgh on the construction of a small World Wide Web-based interactive learning environment (EPIC) developed for the teaching of high performance computing. The paper begins by outlining work done in cognitive science on how people make use of structure in physical environments. Within this context, the EPIC system is presented as an illustration of how the principles can be applied to aid the learning process. User feedback and future plans are also discussed. (Author/DLS)

**********
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
**********

# EPIC : Building a Structured Learning Environment

ED 427 677

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Martin D. Westhead
Edinburgh Parallel Computing Centre
University of Edinburgh
U.K.
martinwe@epcc.ed.ac.uk

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
G.H. Marks
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)"

**Abstract**

*Organisation and structure in the space around us are used to reduce the amount of memory and decision making that must be applied to carry out tasks in everyday life. This paper outlines work in progress at the University of Edinburgh on the construction of a small web based interactive learning environment (EPIC) developed for the teaching of high performance computing. The paper begins by outlining work done in Cognitive Science on how people make use of structure in physical environments, within this context the EPIC system is presented as an illustration of how the principles can be applied to aid the learning process.*

## 1. Introduction

Recent work in Cognitive Psychology [Kirsh95] demonstrates how important the structuring of environments can be in carrying out tasks. The work draws on studies of everyday observations in kitchens, supermarkets and workshop and shows how, by structuring their environment, people simplify the reasoning, memory and perception which are required.

Learning is an activity which is very demanding in terms of cognitive load. It is proposed that the structuring of a learning environment can alleviate unnecessary aspects of that load, and allow the learner to focus on the material being presented. By way of illustration a case study is presented of work carried out at the Edinburgh Parallel Computing Centre (EPCC), on a Web-based learning environment. The system, called EPIC (EPcc Interactive Courseware), extends the hypertext environment of the Web to provide an simple, consistent interface to exercises on a number of courses.

## 2. Structuring Environments

In his paper on intelligent use of space David Kirsh [Kirsh95], develops the beginnings of a principle classification of the way in which people make use of features of the space around them to simplify tasks. We are embodied agents and the nature of our interaction with the world about us is inherently limited. We face in one direction at a time and we can only immediately manipulate objects within our reach. The space about us also conforms to physical and geometric laws. We use these constraints to our advantage and in this section we look at some of the ways in which this is done. Kirsh argues that whether we are aware of it or not, we are constantly organising and reorganising our environments to enhance our performance.

Some of the techniques we use are obvious, but others are learned, and developed over time. Experts in particular excel at organising their tools and materials so as to minimise the number of decisions which they have to have to make in performing a task. For example a short order cook given an order for say, mushroom omelet and toast, will begin by laying out their workspace. Putting the eggs by a bowl with a fork, laying out the mushrooms and bread by a chopping board. Thus in the cooking stage, activity can progress from one item to another at high speed without worrying about the higher level planning involved. Kirsh reports that in cases where several orders are being prepared at the same time the cooks will cluster materials for orders together and leave utensils by ingredients to mark their purpose.

Even in everyday tasks people use spatial ordering to cut down the performance of the task to a series of

IR019196

unreflective actions, triggered by environmental cues. Making a cup of tea in your own kitchen will proceed automatically in a skill based mode of activity. You don't need to think about where the cups, tea kettle are, or how to arrange them. Simple problems are solved with "problem solving packets" determined by rules. Finding that there are no clean mugs you will "automatically" go and wash one. It is only when the task gets out of control that conscious analytical thought needs to be applied.

In order to increase the amount of time a person spends in skill or rule based modes, they can either broaden their range of skills and rules, or they can structure their environment to limit the uncertainty. Such structures can involve physically limiting options or providing perceptual cues to act as reminders. Greying out items on a pull down menu which are not available for selection is one example, another might be placing a library book by the door, as a reminder to return it when you go out. If such cues are successful they capture our attention at critical times and either alert us to opportunities or reduce the likelihood of "double-capture slip", this occurs when we are distracted from a course of action and habit takes over. e.g. "I intended to stop on the way to work to buy shoes, but 'woke up' to find I had driven right past."[Reason82,90]

We will notice that an action is available in the current situation if the environment *affords* that action. An affordance in a particular situation can be described as the way a situation lends itself to being used [Gibson77]. Thus a button affords pushing, a bicycle affords riding, an empty glass affords filling. The affordance offered by a knife and chopping board next to unchopped mushrooms by our fast cook chief is clear. It is important to see that the affordances of an object are features of the agent and the environment, if you cannot ride a bicycle, or if it is chained up, it will not afford riding. In structuring our environment we both hide and highlight affordances to make to change the likelihood of them being chosen.

The structure of an environment is not always the result of an individual's organisation, structure may also be imposed. A kitchen on its own has a great deal of physical structure associated with its use, large sinks, cookers, work surfaces, cupboards etc. Another good example of imposed structure is an assembly line, where efficiency is increased by limiting the tasks an individual performs, and ordering the tasks in a linear fashion.

In summary we exploit the constraints we face as embodied agents interacting with the world about us to simplify the execution of tasks. The next section looks at the EPIC package to illustrate how we can begin to use these principles in a learning environment.

# 3. EPIC

EPIC is a general purpose interactive learning package developed at EPCC. As a leading centre for High Performance Computing (HPC) in Europe, EPCC has a number of different education and training requirements. Within the centre, the education and training group is responsible for the training of staff, users of our facilities, visiting researchers and industrialists, summer students, as well as a wider remit to bring HPC to Higher Education Institutes throughout Scotland and Northern England. With such diverse requirements we are involved not only in classroom based courses, but also in stand-alone course material, and distance learning.

## 3.1 The EPIC system

Many of the courses taught at the centre focus on learning programming languages. This involves the users iterating through a cycle of editing their code, compiling it and running it, returning to the editing stage when things fail. Since this edit-compile-run cycle was at the heart of many of the courses we taught, it seemed like an obvious place to start providing interaction. The World Wide Web by its ubiquity and high quality, free, GUI supporting software was an obvious choice of platform, particularly with distance learning being one of our goals.

Unfortunately the Web was not really designed for interactive courseware. One of the obvious drawbacks being that service is provided on a document, by document basis and no record of a user is

maintained. This `lack of state' makes it difficult to maintain an interaction. There are a number of mechanisms that can be used to achieve the desired result. For example state can be maintained in forms which interact with smart CGI scripts, however we felt that a forms based interface was too limiting. Instead we made use of the external viewer mechanism in most browsers, which causes the browser to pass a page on to an application, if it is of a defined MIME type. We defined a new MIME type for EPIC files will cause a browser to execute a script on the client. Using this mechanism we can create exercise pages with buttons that launch processes on the client machine.

The buttons on an typical exercise page enable the user to:

- launch an editor of their choice, containing the code skeleton associated with the current exercise,
- fire up an interactive shell window which automatically compiles the appropriate file,
- run the executable produced in the same window,
- view the results using a visualisation package,
- clear up all the windows,
- provide configuration facilities, such as changing editor preferences, or number of processors in a run,
- and finally amongst the security measures, is a monitor window which reports all the actions EPIC takes. The actions are strictly limited by the script, but this ensures that the scripts are carrying out the action the user expects.

The EPIC exercises are embedded within a complete set of hypertext course-notes. Instructions to users on commands or techniques in the course can be linked back to the place in the notes where they were first encountered. Links can also be made to other relevant online material such as language specifications, or FAQs about the course.

### 3.2 EPIC as a Structured Learning Environment

Although it is cute to be able to run the compiler from a button on the exercise page, it could be argued that the EPIC package is in fact doing nothing new. Students on a course could do all the things that EPIC allows them to do just by typing in the right command line. However by the same argument the Web itself does nothing new, long before the Web we had anonymous ftp sites. You could go to a site an download documents, pictures, audio, video etc. However the Web makes the whole process much less painful. It provides structure to cyberspace. It relieves the user of the burden of logging on, finding the remote file, copying it somewhere sensible in his or her filespace uncompressing it, unarchiving it, displaying it, and then once read potentially deleting it. The Web presents links which afford clicking, in a context which provides cues to guide browsing, all within an environment with a consistent `look and feel'. Although it is true to say that the Web does nothing that couldn't be done before, it is also true that the cognitive burden which is relieved by this structure, qualitatively changes the experience of retrieving files from the Internet (bandwidth notwithstanding). EPIC extends the structure made possible by the Web technology to apply it to the edit-compile-run cycle of the programming courses. In designing the system, we looked at the various tasks that a user would have to perform, and looked at how an interface could be designed to hide unnecessary details and provide affordances for these tasks. In our case the principle subtasks that we were concerned with were editing, compilation and execution. Before EPIC was introduced courses required users to copy and manage various skeleton code and configuration files that were used though the course and correctly invoke the compiler, execution mechanism, and visualisation packages. This involved remembering, or frequently looking up:

- pathnames,
- filenames,
- compiler commands,
- compiler flags,
- execution commands and
- execution flags.

Furthermore many of these details can change subtly between stages of an exercise. Although this may not sound very significant it does present a burden that someone learning a new programming language

could well do without. All these details are hidden in EPIC by a button push allowing the users to focus on the exercise itself.

EPIC also seems to reduce the likelihood of "double-capture slips". It is frequently the case that you would see a user in a training course come out of an editor, or finish a compilation, and forget what they intended to do next. Habitual response takes over in these cases. In UNIX users this is often to type 'ls' and list the files in the current directory. This behaviour is quite common. In a sense the user is searching for environmental cues that will trigger them to remember the next step. In EPIC the buttons are there on the exercise page, *affording* clicking. The interface is all contained on the Web browser in a consistent style. What is more the buttons are arranged in the expected order of use "edit", "compile", "run", "view","clear-up". Since we naturally read from left to right (most course participants are from the West) the user will encounter the buttons they need in the order they will use them. Admittedly it would be better if buttons were shaded until they could be used, "run" was grayed out until "compile" had been used. Unfortunately the stateless nature of the Web interface, and the tenuous nature of the communication with the client script make this difficult to achieve.

Finally EPIC provides a consistent `look and feel' to the EPCC courses. Some of our users are unfamiliar with the Web, the interface is sufficiently simple that they quickly get the hang of using it. Since the same interface is used on all the exercises, across different courses the user need only learn how to use it once.

### 3.3 Use of EPIC

EPIC has been in use for over a year now. We currently have EPIC courses in MPI (Message Passing Interface) and HPF (High Performance Fortran) and the system is being routinely introduced for our classroom courses on these topics. A great deal of interest has been shown by a number of people in the software, which is now publicly available, and courses have been run using EPIC at Glasgow and Aston Universities.

EPIC has been used within the centre on around 30 users, in roughly half and half stand-alone, and classroom based courses. This represents in the region of 200 hours of use. Feedback from the users has been obtained in the form of questionnaires, and this information has been used to guide the development of the system.

On the whole feedback has been very positive. When asked what he most liked about the course, one user wrote:

*"The possibility, with EPIC, to concentrate on the main parts of the development of the programs."*
Other comments include:

*"...one can concentrate on really learning MPI, without wasting time on compiler options."*

*"I was unable to attend an MPI course given by EPCC staff and had an urgent need to learn the basic MPI routines. This course enabled me to do so quickly and at my own pace."*

On general course questionnaires EPIC was frequently listed as a positive feature.

Of course there were criticisms too. The response speed of the system was too slow for some. We also had two experienced programmers who found the exercise styles and interface too restrictive. Other training groups have commented that they feel that some of the details which EPIC hides it would be useful to learn, compiler flags for example.

We feel quite strongly that for our courses, these details are superfluous to what is being taught. There are many language compilers, each with different execution commands, and different flags. So in learning these details a user would only really benefit if they were using exactly the same system as EPCC. Even then these are hard to remember and are easily looked up.

In the past users unfamiliar with UNIX had to learn how to get around a new operating system in order take a course. EPIC abstracts some or all of these details away so that the course is really about the material it sets out to teach, not the vagaries of the operating system or compiler that is being used.

### 3.4 Future plans

The EPIC project has been about developing a working tool that we can use in the Centre. The development has involved extensive field testing and user feedback to ensure that we were building something useful. Despite the positive feedback so far the current implementation is only a small step towards what could be accomplished in providing a learning environment. However this is work in progress and we have a number of plans to build on this initial work.

The EPIC system has just reached the end of its first phase of development (at time of writing), the software has been made publicly available, and we are encouraging people to download the client and try the courses. However the project is ongoing, and there are a number of future directions in which we hope to take the system. The most significant developments will be towards improving the system's potential for distance learning. We envisage at least three possibilities:

1. Currently it is necessary for distance users to have their own copies of the compilers and other software used. This limits the distance audience. A solution would be to provide another interface which appears almost identical to the user but which causes programs to be compiled and run at EPCC. There are of course security issues associated with this, but these can be solved.
2. Interaction in the current implementation is simply interaction with a machine. It is important in the creation of a rich learning environment to extend this to facilitate interaction with other people. There are two directions we want to take this in.
    - Communication with an on-line tutor --- A user would hit a button to connect with a tutor, this would transfer log information to the tutor automatically setting up a mirror of the students page and code, and creating a chat window, or frame within the browser to allow the students to communicate directly with an expert advisor.
    - Communication between students --- We hope to set up bulletin board and mailing list to build on the experiences of people who have taken the course. This will provide another feedback mechanism to help us tune the course, and encourage the students to help each other in the learning process.
3. Taking the communication a stage further we are also looking at the possibility of using the MBone to transmit interactive courses around Scotland. This would give us video conferencing facilities, which would complement the EPIC tutoring facilities. In a sense the EPIC system would allow the tutor the electronically look over the shoulder of a student at a remote site.

The mechanism that the system uses is very general, and could be used to activate a number of other packages and applications on the user's machine. We are already developing two new courses that will use EPIC to drive applications in this way. Other include the possibilities presented by combining Java applets with EPIC, to increase the `intelligence' available at the page.

## 4. Conclusion

It has been argued that the principles of structuring the environment can be used to hide unnecessary detail from the student in a learning environment. EPIC represents an application of this principle. Although we haven't carried out any controlled studies on its effect of learning, the anecdotal evidence we have received from questionnaires suggests that the support it provides is helping.

## Acknowledgements

Work on the EPIC system and courses has involved a number of staff at EPCC, in particular Alistair Ewing and Mario Antonioletti whose efforts have been invaluable.

# References

[Kirsh95] David Kirsh. "Intelligent use of Space". Artificial Intelligence 73 (1995) pp31-68.

[Reason82] J.T. Reason and K. Mycielska, "Absent-Minded? The Psychology of Mental Lapses in Everyday Errors" Prentice-Hall, Englewood Cliffs, NJ, 1982.

[Reason90] J.T. Reason "Human Error". Cambridge University Press, Cambridge, England, (1990).

[Gibson77] J. J. Gibson. "The Theory of Affordances", in: R. E. Shaw and J. Bransford eds. "Perceiving, Acting and Knowing" Lawrence Erlbaum, Hillsdale NJ, 1977. Note: this paper is currently under consideration for WebNet'96